

**Amendment to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A computer system for selectively retrieving runtime objects in an application development environment, comprising:

a first memory unit storing a plurality of server runtime objects; and

a second memory unit storing a plurality of local runtime objects, each local runtime object including a generation setting set prior to the respective local runtime object's generation and indicating a manner according to which ~~associated with generation of the~~ respective local runtime ~~object~~ object's generation was to have been conducted; and

a processor responsive to a request for a requested runtime object by being configured to retrieve a valid copy of the requested runtime object from the plurality of local runtime objects if therein, and to otherwise retrieve the valid copy of the requested runtime object from the plurality of server runtime objects if therein;

wherein the processor is configured to invalidate a local runtime object when the local runtime object's generation setting does not match a current generation setting.

Claim 2. (Canceled).

3. (Previously Presented) The system of claim 1, wherein the processor is configured to retrieve the local runtime object by returning a data element indicating the requested local runtime object's validity.

4. (Original) The system of claim 1, further comprising a local database including a first data structure and a second data structure, the second data structure configured to store a plurality of pointers, at least one pointer configured to identify a local runtime object from the plurality of local runtime objects, the first data structure configured to store a plurality of commands, the commands configured to manipulate the second data structure.

5. (Previously Presented) A computer system for selectively retrieving runtime objects in an application development environment, comprising:

- a server database configured to store a plurality of server runtime objects;
- a local database configured to store a plurality of local runtime objects; and
- a generator component configured to include a first generator and a second generator, each provided in communication with each other and the server and local databases,

- the generator component configured to invalidate and validate server and local runtime objects, to retrieve a requested one of the server and local runtime objects, and to regenerate the requested runtime object conditional upon the retrieved runtime object's invalidity,

- the first generator configured to perform a first portion of the generator component's tasks, the second generator configured to perform a second portion of the generator component's tasks.

6. (Original) A computer system for retrieving stored runtime objects in an application development environment, comprising:

- a local database to store a plurality of pointers and a plurality of local runtime objects, at least one local runtime object from the plurality of local runtime objects including a content, a state, and an original checksum attribute,

- the original checksum attribute configured to represent a combination of the content and the state of the local runtime object with which the original checksum attribute is associated,

- at least one pointer configured to identify, from the plurality of local runtime objects, a local runtime object including an original checksum attribute,

- the at least one pointer configured to include a copy of the original checksum attribute associated with the local runtime object that the at least one pointer is configured to identify; and

- a generator component to, in response to a request for a local runtime object:

- calculate a new checksum attribute associated with the requested local runtime object,

- compare the requested local runtime object's new checksum attribute to its pointer's copy of the original checksum attribute,

invalidate the requested local runtime object when the new checksum attribute and the copy of the original checksum attribute do not match, and

retrieve the requested local runtime object if the requested local runtime object remains valid.

7. (Previously Presented) A computer system for selectively retrieving runtime objects in an application development environment, comprising:

a server database configured to store a plurality of server runtime objects;

a local database configured to store a plurality of local runtime objects, each local runtime object from the plurality of local runtime objects configured to correspond to one server runtime object from the plurality of server runtime objects; and

a generator component responsive to a request for a requested runtime object by being configured to:

determine if the plurality of server runtime objects includes a valid copy of the requested runtime object;

if it is determined that the plurality of server runtime objects includes the copy of the requested runtime object, determine if the plurality of local runtime objects includes a runtime object that corresponds to the valid copy of the requested runtime object; and

retrieve the requested runtime object from the plurality of local runtime objects if it is determined that the plurality of local runtime objects includes the runtime object that corresponds to the valid copy of the requested runtime object, and to retrieve the valid copy of the requested runtime object from the plurality of server runtime objects otherwise, if therein.

8. (Original) The system of claim 7,

wherein at least one local runtime object from the plurality of local runtime objects and at least one server runtime object from the plurality of server runtime objects contain a content attributes and a state attributes, and

wherein the each local runtime object corresponds to the one server runtime object when the each local runtime object's content attributes and state attributes match the one server runtime object's content attributes and state attributes.

9. (Currently Amended) A computer-implemented method for selectively retrieving runtime objects in an application development environment, comprising:

storing a plurality of server runtime objects;

storing a plurality of local runtime objects, each local runtime object including a generation setting set prior to the respective local runtime object's generation and indicating a manner according to which ~~associated with generation of~~ the respective local runtime object ~~object's generation was to have been conducted~~; ~~[[and]]~~

responding to a request for a requested runtime object by retrieving a valid copy of the requested runtime object from the plurality of local runtime objects if therein, and otherwise by retrieving the valid copy of the requested runtime object from the plurality of server runtime objects if therein; and

invalidating a local runtime object when the local runtime object's generation setting does not match a current generation setting.

Claim 10. (Canceled).

11. (Original) The method of claim 9, wherein the local runtime object is retrieved by returning a data element indicating the requested local runtime object's validity.

12. (Original) The method of claim 9, further comprising:

storing a plurality of pointers, at least one pointer identifying a local runtime object from the plurality of local runtime objects; and

storing a plurality of commands, the commands manipulating the plurality of pointers.

13. (Original) A computer-implemented method for retrieving runtime objects in an application development environment, comprising:

storing a plurality of pointers;

storing a plurality of local runtime objects, at least one local runtime object from the plurality of local runtime objects including a content, a state, and an original checksum attribute,

the original checksum attribute representing a combination of the content and the state of the local runtime object with which the original checksum attribute is associated,

at least one pointer identifying, from the plurality of local runtime objects, a local runtime object including an original checksum attribute,

the at least one pointer including a copy of the original checksum attribute associated with the local runtime object that the at least one pointer identifies; and

responding to a request for a local runtime object by:

calculating a new checksum attribute associated with the requested local runtime object,

comparing the requested local runtime object's new checksum attribute to its pointer's copy of the original checksum attribute,

invalidating the requested local runtime object when the new checksum attribute and the copy of the original checksum attribute do not match, and

retrieving the requested local runtime object if the requested local runtime object remains valid.

14. (Previously Presented) A computer-implemented method for selectively retrieving runtime objects in an application development environment, comprising:

- storing a plurality of server runtime objects;
- storing a plurality of local runtime objects, each local runtime object from the plurality of local runtime objects capable of corresponding to one server runtime object from the plurality of server runtime objects; and
- responding to a request for a requested runtime object by:
  - determining if the plurality of server runtime objects includes a valid copy of the requested runtime object;
  - if it is determined that the plurality of server runtime objects includes the copy of the requested runtime object, determining if the plurality of local runtime objects includes a runtime object that corresponds to the valid copy of the requested runtime object; and
  - retrieving the requested runtime object from the plurality of local runtime objects if it is determined that the plurality of local runtime objects includes the runtime object that corresponds to the valid copy of the requested runtime object, and by retrieving the valid copy of the requested runtime object from the plurality of server runtime objects otherwise, if therein.

15. (Original) The method of claim 14,

- wherein at least one local runtime object from the plurality of local runtime objects and at least one server runtime object from the plurality of server runtime objects contain a content attributes and a state attributes, and
- wherein the each local runtime object corresponds to the one server runtime object when the each local runtime object's content attributes and state attributes match the one server runtime object's content attributes and state attributes.

16. (Previously Presented) A computer system for selectively retrieving runtime objects in an application development environment, comprising:

- a first memory unit storing a plurality of server runtime objects; and
- a second memory unit storing a plurality of local runtime objects generated based on corresponding server runtime objects, each local runtime object including a respective identification of one of a plurality of generation settings, the identified generation setting being associated with the generation of the respective local runtime object; and
- a processor configured to respond to a request for a requested runtime object by retrieving a valid copy of the requested runtime object from the plurality of local runtime objects if therein, and to otherwise retrieve the valid copy of the requested runtime object from the plurality of server runtime objects if therein.

17. (Withdrawn) The system of claim 16, wherein, based on a particular development runtime object, the processor is configured to generate different local runtime objects depending on which one of the plurality of generation settings is enabled.

18. (Withdrawn) The system of claim 17, wherein the plurality of generation settings are user-entered generation settings.

19. (Withdrawn) The system of claim 17, wherein the processor, responsive to the request, is configured to:

- (a) determine if a copy of the requested runtime object is included in the pluralities of runtime objects;
- (b) if the processor determines in (a) that the copy is included in the pluralities of runtime objects, compare the generation setting identified by the identification included in the copy with a currently enabled generation setting; and
- (c) determine, based on the comparison, whether the copy is valid, the copy being considered valid conditional upon the identified generation setting of the copy matching the currently enabled generation setting.

20. (Withdrawn) The system of claim 16, wherein:  
for different ones of a plurality of development object types, different runtime object types are generated; and

the generation settings are settings applied universally for generation of runtime objects corresponding to any of the plurality of different development object types.

21. (Withdrawn) The system of claim 5, wherein, the generator component is configured to respond to a request for a runtime object that reflects a user's inputted development object by:

comparing a generation timestamp included in a particular server runtime object that corresponds to the development object with a modification timestamp included in the development object;

comparing a generation timestamp included in a particular local runtime object that corresponds to the particular server runtime object with the generation timestamp included in the particular server runtime object;

comparing a generation setting included in one of the particular server and local runtime objects with a currently enabled generation setting;

performing a checksum validation upon the one of the particular server and local runtime objects;

determining that the one of the particular server and local runtime objects is valid conditional at least upon (a) its generation setting matching the currently enabled generation setting, (b) a time indicated by its generation timestamp being one of equal to and later than a time indicated by the generation timestamp of the development object, and (c) a validity result of the checksum validation;

returning, for use by the user, the one of the particular server and local runtime objects if it is determined that the one of the particular server and local runtime objects is valid; and

if the generator component one of (i) does not find any previously generated runtime objects that corresponds to the development object and (ii) determines that all previously generated runtime objects that correspond to the development object are invalid:

generating a new runtime object; and

returning the new runtime object for use by the user.